

问题 第 5 周 2.1：寻墙系统功能

使用 `sf.R`、`sf.Gain`、`sf.Cascade`、`sf.FeedbackAdd` 和 `sf.FeedbackSubtract` 来构建寻墙机器人系统函数。这些过程在软件文档中的 `sf` 模块中有文档说明。

定义一个名为 `controllerSF` 的 Python 程序，它接受一个参数：

- 应用于误差的比例增益 k 。

并且它返回一个系统的系统函数，该系统的输入是距离误差，输出是指令速度。

定义一个名为 `plantsF` 的 Python 程序，它接受一个参数：

- 时间步长 T 。

并且它为一个系统返回一个系统函数，该系统的输入是给定的速度，输出是到墙壁的实际距离。

定义一个名为 `sensorsE` 的 Python 程序，它不带有任何参数，并返回一个系统函数，该系统的输入是实际传感器值，输出是一步延迟后的传感器读数。

定义一个名为 `wallFinderSystemsF` 的 Python 程序，它接受两个参数：

- 时间步长 T 。
- 应用于误差的比例增益 k 。

并且它返回一个系统函数，用于这样一个系统，其输入是期望的距离，输出是到墙壁的实际距离。

您可以在 `Idle` 中调试这些内容，方法是将您的定义放入一个文件中，并在顶部添加“`import lib601.sf as sf`”。

您可以通过将获得的系统函数与您在上一个设计实验问题中推导出的各种系统的差分方程进行比较，来检查是否得到了正确的答案。您需要差分方程转换为系统函数（反之亦然）来进行比较——要小心系数的顺序和负号……

```
def controllerSF(k):  
    pass  
  
def plantSF(T):  
    pass  
  
def sensorSF():  
    pass  
  
def wallFinderSystemSF(T, k):  
    pass
```

问题 第 5 周 3.2：壁面跟随系统功能

输入比例跟踪器的系统函数，将分子和分母多项式以 R 语言中的表达式形式输入。输入多项式的方式应类似于在 Python 中输入表达式的方式（尽管在正常的代数记法中省略 * 应该也能工作）。您可以使用以下符号：

R, T, V, k

(这些变量是区分大小写的)

以这种格式输入的 R 中多项式的一个示例是：

`k*T*(R**2) - V*(T**2)*R + 1`

- 在检查之前，您需要同时输入分子和分母。它们将一起被检查。检查的是比率是否正确，而不是各个部分是否正确。
- 大量使用括号以避免歧义。

“分子：”

分母：

问题 第 5 周 3.4 节：壁面跟随模型

使用 `sf.R`、`sf.Gain`、`sf.Cascade`、`sf.FeedbackAdd` 和 `sf.FeedbackSubtract` 来构建寻墙机器人系统函数。这些过程在软件文档中的 `sf` 模块中有文档说明。

定义一个名为 `wallFollowerModel` 的 Python 程序，它接受三个参数：

- 应用于误差的比例增益 k 。
- 时间步长 T
- 平移速度 v

并且它返回一个系统函数，用于这样一个系统，其输入是期望的距离，输出是到墙壁的实际距离。

您可以在 `Idle` 中调试这些内容，在 `designLabo5work.py` 文件中。

